

Actividad Jenkins

Ficha del Informe

Autor: Sergio Martín Santana

Nombre: Actividad Jenkins

Asignatura: Laboratorio y Desarrollo de Herramientas.

Entrega: 06/12/2015

Descripción:

Haga un análisis con Jenkins del código desarrollado por usted en la actividad en clase (Algoritmo de distancia de Levenshtein). El código además de las pruebas incluidas en clase debe de estar comentado para generar la documentación con javadoc. Recuerde que el código creado en clase debe de estar desarrollado como un proyecto Maven, lo que facilitaría su despliegue y análisis en Jenkins. El código debe de estar subido en un sistema de control de versiones (GitHub, Git, o Bitbucket) desde el cual se realizará el análisis con el Jenkins.

Enlace al código fuente, [aquí](#).

Configurando Jenkins

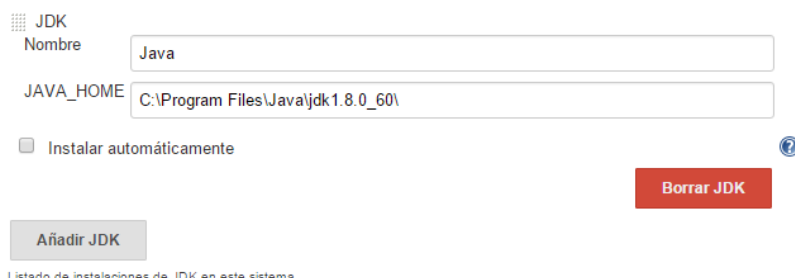
Lo primero para configurar el sistema Jenkins será descargarlo desde su página oficial gratuitamente, seleccionando la versión de nuestro ordenador, tanto la plataforma en la que correrá como la arquitectura con la que cuenta.

Una vez descargado lo instalaremos y empezaremos con su configuración.

Abriendo cualquier navegador web e introduciendo en la barra de direcciones, la siguiente ruta <http://127.0.0.1:8080/> y clicando en Administrar Jenkins/Configurar el Sistema, entraremos en el panel de configuración. En primer lugar realizaremos la configuración del entorno Java, para ello podremos decidir que Jenkins descargue automáticamente el SDK, desde la página oficial de Sun, dándonos de alta en sus página web, o por el contrario podemos utilizar una versión que tengamos nosotros instalada en nuestro equipo, en nuestro caso utilizaremos este camino.

JDK

instalaciones de JDK



JDK

Nombre

JAVA_HOME

☐ Instalar automáticamente

Listado de instalaciones de JDK en este sistema

Lo siguiente será configurar el sistema Maven , el cual al no tenerlo instalado en nuestro equipo de prueba, nos hemos decantado por hacer que se instale automáticamente por el Sistema de Integración Continua Jenkins.

Maven

instalaciones de Maven

Maven

Nombre Maven

☒ Instalar automáticamente

Instalar desde Apache

Versión 3.3.9

Borrar un instalador

Añadir un instalador

Añadir Maven

Listado de instalaciones de Maven en este sistema

Y a continuación aparecen las configuraciones de Maven, las cuales dejaremos de manera predeterminada.

Por último y para terminar, configuraremos Git en Jenkins. Git no viene por defecto, así que lo que tenemos que hacer es instalar el plugin de Git para Jenkins. Para ello, volvemos a pulsar en el menú “Administrar Jenkins” y pulsamos en la opción “Administrar Plugins”. Se nos mostrará la información de los plugins, pulsando en la pestaña “Todos los plugins”, nos aparece una lista con todos los plugins que podemos instalar, así que filtramos por “git plugin” y lo seleccionamos para su instalación, y pulsamos el botón “Instalar sin reiniciar”. Una vez instalado reiniciamos Jenkins para que los cambios tengan efecto.

Configurando el Proyecto LDH_Distancias.

Para empezar a utilizar esta herramientas deberemos de crear una nueva tarea en Jenkins, de la siguiente forma. Primeramente añadiremos un nombre y tipo de repositorio en la primera ventana que nos mostrará Jenkins.

Jenkins

Nombre de la Tarea LDH_Distancia

Crear un proyecto de estilo libre

Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.

Crear un proyecto maven

Ejecuta un proyecto maven. Jenkins es capaz de aprovechar la configuración presente en los ficheros POM, reduciendo drásticamente la configuración.

Crear un proyecto multi-configuración

Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en múltiples entornos, ejecutar sobre plataformas concretas, etc.

External Job

Este tipo de tareas te permite registrar la ejecución de un proceso externo a Jenkins, incluso en una máquina remota. Está diseñado para usar Jenkins como un panel de control de tu sistema de automatización. Para más información consulta esta [página](#).

Copiar una Tarea existente

Copiar desde

OK

Lo siguiente será configurar el resto de campos para conseguir que Jenkins ejecute el código automáticamente, así como las pruebas. Para ello iremos destacando los campos modificados para su configuración.

Lo primero será configurar el origen del código fuente y de las pruebas.

Configurar el origen del código fuente

☐ Ninguno
☐ CVS
☐ CVS Projectset
☒ Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

Navegador del repositorio

Después configuraremos cada cuanto se dispararán las ejecuciones del código anterior. En nuestro caso se hará periódicamente, cada 15 minutos, pero podría ser seleccionada cualquiera de las otras dos opciones, además de que cuenta con un apartado de ayuda muy completo, para entender las opciones.

Disparadores de ejecuciones

☐ Build after other projects are built
☐ Consultar repositorio (SCM)
☒ Ejecutar periódicamente

Programador

Would last have run at domingo 6 de diciembre de 2015 17H10' GMT; would next run at domingo 6 de diciembre de 2015 17H25' GMT.

También indicaremos las tareas que pretendemos que Maven realice, es decir que “Goals” definiremos.

- **validate:** Valida que el proyecto es válido y contiene los elementos necesarios
- **compile:** Compila el código
- **test:** Si utiliza un “unit testing framework” lo ejecuta. Estos test no necesitan desplegar la aplicación
- **package:** Compila y empaqueta el código (como jar, etc.)

Aunque hayamos indicado el “goal” package, realizará todos los que aparecen con anterioridad, como podrían ser los Test. Aunque podríamos indicar que solo queremos realizar los test, añadiendo como paso, TEST.

Ejecutar

Ejecutar tareas 'maven' de nivel superior

Version de Maven: (Por defecto)

Goals: clean package

Avanzado...

Borrar

Añadir un nuevo paso

Una vez configurado, Jenkins se ejecutará siguiendo lo expuesto en campo que determinamos con anterioridad, en nuestro caso cada 15 minutos. En nuestro caso, después de seguir las instrucciones en la configuración existen algunos problemas en la conexión al código fuente al parecer. Por lo que las pruebas nos aparecen Fallidas.

Jenkins

Jenkins > LDH_Distancias

Proyecto LDH_Distancias

Volver al Panel de Control

Estado Actual

Cambios

Zona de Trabajo

Construir ahora

Borrar Proyecto

Configurar

Historia de tareas

Tendencia

find

#	Fecha
#7	06-dic-2015 17:29
#6	06-dic-2015 17:25
#5	06-dic-2015 17:22
#4	06-dic-2015 17:13
#3	06-dic-2015 17:12
#2	06-dic-2015 17:10
#1	06-dic-2015 17:09

RSS Para Todos RSS para los errores

Enlaces permanentes

- Última ejecución (#7) hace 1 Min 2 Seg
- Última ejecución fallida (#7) hace 1 Min 2 Seg
- Última ejecución fallida (#7) hace 1 Min 2 Seg
- Last completed build (#7) hace 1 Min 2 Seg

Desactivar el Proyecto

Podremos, clicando en cada una de las ejecuciones, ver el estado en que concluyó la misma. Donde se nos indicará el nombre así como la fecha en la que se realizó la prueba.

Jenkins

Jenkins > LDH_Distancias > #7

Compilar #7 (06-dic-2015 17:29:41)

Comenzó hace 2 Min 37 Seg

Tardó 0.21 Seg

añadir descripción

Volver al proyecto

Estatus

Cambios

Console Output

Editar información de la ejecución

Borrar Proyecto

Git Build Data

Ejecución previa

Imposible de determinar (log)

Iniciada por usuario anonymous

git Revision:

También podremos visualizar el estado del Log, para consultar la ejecución, pinchando en el enlace que aparece al final del texto junto al primer icono del panel central.

The screenshot shows the Jenkins web interface. On the left, a sidebar contains links like 'Volver al proyecto', 'Estatus', 'Cambios', 'Console Output', 'View as plain text', 'Editar información de la ejecución', 'Borrar Proyecto', 'Git Build Data', and 'Ejecución previa'. The main area is titled 'Salida de consola' and displays the following log text:

```
Lanzada por el usuario anonymous
Ejecutando en el espacio de trabajo C:\Program Files (x86)\Jenkins\workspace\LDH_Distancias
Cloning the remote Git repository
Cloning repository git@github.com:hirosms/LDH.git
> C:\Program Files\Git\git-cmd.exe init C:\Program Files (x86)\Jenkins\workspace\LDH_Distancias # timeout=10
Fetching upstream changes from git@github.com:hirosms/LDH.git
> C:\Program Files\Git\git-cmd.exe --version # timeout=10
> C:\Program Files\Git\git-cmd.exe fetch --tags git@github.com:hirosms/LDH.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\git-cmd.exe config remote.origin.url git@github.com:hirosms/LDH.git # timeout=10
> C:\Program Files\Git\git-cmd.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\git-cmd.exe config remote.origin.url git@github.com:hirosms/LDH.git # timeout=10
Fetching upstream changes from git@github.com:hirosms/LDH.git
> C:\Program Files\Git\git-cmd.exe --version # timeout=10
> C:\Program Files\Git\git-cmd.exe --version # timeout=10
> C:\Program Files\Git\git-cmd.exe fetch --tags git@github.com:hirosms/LDH.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\git-cmd.exe rev-parse "origin/master^{commit}" # timeout=10
> C:\Program Files\Git\git-cmd.exe rev-parse "master^{commit}" # timeout=10
ERROR: Couldn't find any revision to build. Verify the repository and branch configuration for this job.
Finished: FAILURE
```

Volviendo al panel de control de Jenkins, nos podremos comprobar el estado de la última ejecución de nuestras tareas, así como una estadística de como se ha ido comportando en el tiempo.

The screenshot shows the Jenkins dashboard. On the left, a sidebar contains links like 'Nueva Tarea', 'Personas', 'Historial de trabajos', 'Administrar Jenkins', and 'Credenciales'. The main area displays a table of build jobs. The table has columns: 'S' (Status), 'W' (Workspace), 'Nombre' (Name), 'Último Éxito' (Last Success), 'Último Fallo' (Last Failure), and 'Última Duración' (Last Duration). The table shows one job, 'LDH_Distancias', which is in a failed state (red icon) and has a duration of 0.21 Seg. Below the table, there are links for 'Icono: S M L', 'Guía de iconos', 'RSS para todos', 'RSS para fallos', and 'RSS para los más recientes'.

S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración
		LDH_Distancias	N/D	5 Min 35 Seg - #7	0.21 Seg